Team AUPAIR Technical Paper for RoboCup@Home 2018 Social Standard Platform League

Jinyoung Choi[1], Beom-Jin Lee[2], Chung-Yeon Lee[2], Kibeom Kim[3] and Byoung-Tak Zhang[123]

[1]Interdisciplinary Program in Cognitive Science,
[2]Department of Computer Science ,
[3]Interdisciplinary Program in Neuroscience
Seoul National University, Seoul, Korea
{*jychoi*, *bjlee*, *cylee*, *kbkim*, *btzhang*}@bi.snu.ac.kr
`https://bi.snu.ac.kr/Robocup/2018/index.html`

Abstract. Team AUPAIR aims to develop intelligent mobile cognitive robots with a machine learning. We envision a new paradigm of service robot with state-of-the-art deep learning methods to carry out difficult and complex real world tasks. We propose a novel integrated perception system for service robots which provides an elastic parallel pipeline to integrate multimodal vision modules, including state-of-the- art deep learning models. On top of that, we deploy highly sophisticated modules such as socially-aware navigation, visual question-answering, and schedule learning. We have released the early version of our perception system that won the 1st place in the RoboCup@Home 2017 SSPL and will post updates for this year's challenge in our website.

1 Introduction

It is our belief that deep learning will give rise to a new paradigm of intelligent service robots by providing highly accurate and robust perception/action ability. To develop such intelligent mobile home robots, we integrate several state-of-the-art deep learning modules for vision, language, and action. Some modules are the results of our own research and others are open-sourced models. This allows the robot to potentially solve problems in a complex real world environment. Especially, we demonstrate our researches about seamless integration of modalities such as visual question answering, concept building, and schedule learning from multimodal sensor data.

2 Team AUPAIR

The term AUPAIR is originally from the name for a domestic assistant from a foreign country working for and living as part of the host family. With this originality, team AUPAIR aims to build an intelligent mobile home robot that

learns the objects, people, actions, events, episodes and schedule plans from daily to extended periods of time as a member of the family. We work towards this goal by developing specific technologies with our machine learning architectures which are applicable to multiple robot platforms. Funded by the Air Force Office of Scientific Research, Prof. Byoung-Tak Zhang supervises a team lead by Chung-Yeon Lee to focus on these main goals and integrate multidisciplinary machine learning based applications to the AUPAIR robot.

3 Our Approach : Hardware

Our main robot platform is the Softbank Pepper, which is the standard platform used in SSPL league. We also use high-end external computing server since our team highly depends on GPU computing.

Fig. 1. Softbank Pepper Platform

3.1 Pepper Platform

Pepper is a omni wheel based mobile robot (Figure 1). It has 2 cameras on its face and there is a **Xtion camera sensors** inside the eyes. The robot also have 4 microphones on its head so it is able to detect the direction of sounds. At the bottom, there are 6 laser sensors and 2 ultrasonic sensors for navigation and collision avoidance. On top of that, Pepper have tactile sensors for natural human-robot interactions (HRI). The platform has quad-core CPU and 4GB RAM. However, since our deep learning modules require high-end GPU computing, we connect external computing devices using Wifi.

3.2 External Computing Devices

Since we deploy many state-of-the-art deep learning models, high-end GPU computing is essential for our team. We use a GPU server to carry out tasks such as object detection, object recognition, human pose estimation, human re-identification, image captioning, visual question answering, etc. The server has Intel i7-6700k processor 3.4 GHz, 32GB RAM, Nvidia TitanX Pascal GPU with 12GB memory. We use ROS Indigo to connect the server to the robot with Wifi.

4 Our Approach : Software

We have used various modules for vision, language and action. Especially, we heavily depend on state-of-the-art deep learning models for vision modules.

4.1 Vision Modules

Object Detection We deploy YOLOv2[10] model trained on MSCOCO dataset[9] for its state-of-the-art running speed and performance. From RGBD images, the object detector module extracts bounding boxes and class label as well as the location of the center of the bounding box with regards to the robot or map. Since the YOLOv2 model provides only coarse class labels(e.g. bottle, box) and service robots often need more specific object labels(e.g. beer bottle, drug box), we additionally append the histogram matcher for fine-grained object recognition. The histogram matcher automatically matches the color histogram of detected objects with saved object images of same class and appends tag for fine-grained label.

Person Identification We use improved Siamese convolutional neural network architecture[1] to predict probability whether 2 images are belong to same person or not. We build the mini-batch that contains all combinations of cropped images of people estimated by the object detector module and saved images of people. Then we assign the identity tag with highest score if the highest score is higher than threshold. To avoid assigning two detection instances to same identity, we assign the identity tag with next highest score if there is another detection instance that has higher score to same identity.

Pose Estimation We deploy OpenPose[2] as our pose estimation module. The original model uses 2-branch convolutional neural networks(CNN)[6] for joint detection and association score between detected joints. Then the model performs bipartite matching to assign joints to each individuals. The original model uses 368*654 size image of whole scene to detect joints of all people in the scene. To solve the instance matching problem between object detector and pose estimator, we modified the model to use the cropped image of people detected by the object detector module. We resize each cropped image to 92*164 and build a

mini-batch of all cropped images to process the images of all detected people in parallel. Since individual people are already cropped, we omitted the bipartite matching part of the original model. With this modification, we achieve faster running speed than vanilla model.

Image Captioning We use Densecap[3] as our image captioning module. Densecap extracts ROI bounding boxes from the whole scene and generate a caption for each box. We modified the model in a similar way to pose estimator. Instead of feeding the whole scene, we feed cropped image of detected objects and reduce the number of ROI proposal so the model outputs 5 captions per ROI.

4.2 Speech Recognition

We deploy the Google Cloud Speech Recognition API. The API not only shows state-of-the-art performance but also supports the user to set hints to further increase the accuracy. Together with our integrated perception system, which we will introduce in Section 5.1, hint setting functionality plays large role in our HRI system.

4.3 Navigation

For navigation, we deploy ROS navigation stack (gmapping + amcl + movebase). We also implemented the reflex function that moves robot in the opposite direction of the obstacles when the robot detects the obstacles within few inches.

5 Innovative Technology and Scientific Contribution

In this section, we present the recent results of our researches. We mainly focus on the integration of modalities, which is crucial for service robot to perform complex tasks in real world.

5.1 Integrated Perception Framework

Service robots are required to understand the holistic context of the environment by integrating multimodal sensory information. We developed IPSRO (Integrated Perception for Service RObots) framework, which is ROS-friendly integrated perception system (Figure 2). IPSRO can flexibly integrate several perception modules including deep learning models to extract rich and useful perceptual information from the environment based on a unified perception representation. On top of that, IPSRO can process the generated perceptual information to perform complex perception tasks. The early version of the IPSRO framework, which played crucial role in winning the RoboCup@Home 2017 SSPL, is available at our team's website.

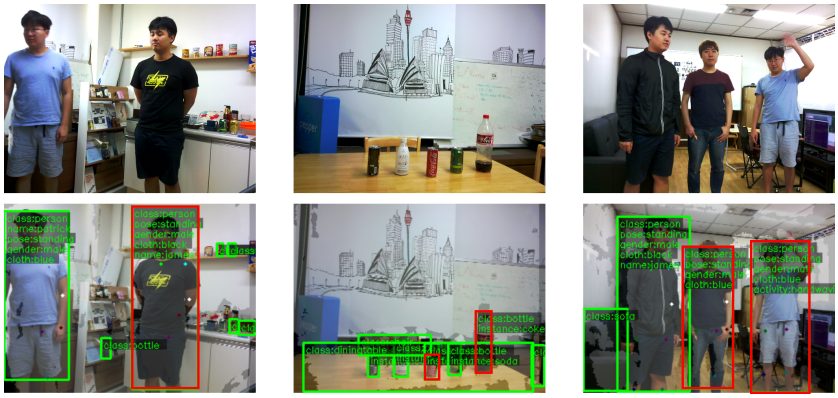

Fig. 2. IPSRO framework

5.2 Robust Human Following by Deep Bayesian Trajectory Prediction

The capability of following a person is important in service robots. Though a vast variety of following systems exist, they lack robustness when the robot losts its target. We developed a robust human following system that uses variational Bayesian techniques for trajectory prediction. Our model accurately predicts the trajectory of the target when target is lost.

5.3 Schedule Learning on Robot Platform

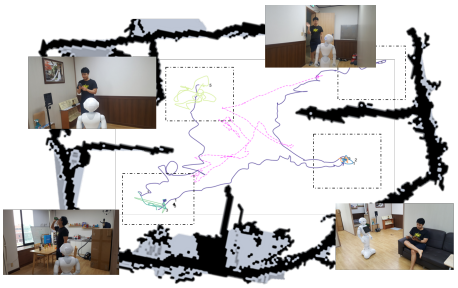

Fig. 3. Schedule learning

A robot needs to understand ongoing events to operate safely and efficiently in complex real situations. The robot needs to predict the current and next events to perform appropriate tasks. Event prediction is generally processed based on the environmental understanding. However, we attempted to perform

this task in advance using pseudo schedule data[7] (Figure 3). The schedule data consists of time, location, action, and event labels. To predict an event from the schedule data vectors, we used a Multiplicative-Gaussian Hypernetworks model[8][5] which is a molecular evolutionary architecture for cognitive learning and memory.

5.4 Visual Conversation Robot for Interactive Engagement

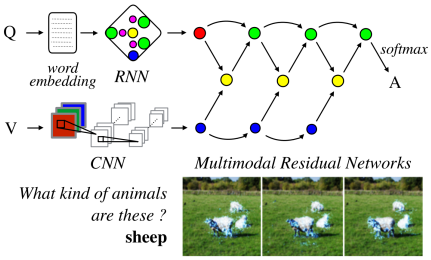

Fig. 4. Visual question answering

To achieve human-level artificial intelligence, it is crucial to develop algorithms which handle human-like visual and linguistic information. We propose the Multimodal Residual Networks (MRN)[4] for the multimodal residual learning in assumption of visual question-answering tasks. It extends the idea of the deep residual learning, which learns joint representation from vision and language information effectively. Using the MRN, our robot system can answer questions about what the robot is seeing (Figure 4).

6 Re-usability of the system

Although we fine-tuned our system to the Pepper platform, We aim our technology to be applicable to any platform that uses ROS. Therefore, our system is mostly applicable for any ROS compatible service robots. On the top of that, we have open sourced our integrated perception framework to encourage the progress of entire service robot R&D community.

7 Applicability of the approach in the real world

Our ultimate goal is to deploy the service robots in real human environment. For this purpose, we do our best to verify our research in realistic home like environment (Figure 5, left). Thanks to our endeavor to develop robust general purpose service robot, we achieved highest score in the complex and realistic arena environment of RoboCup@Home 2017 SSPL (Figure 5, right). The videos of competition can be found in our team's website.

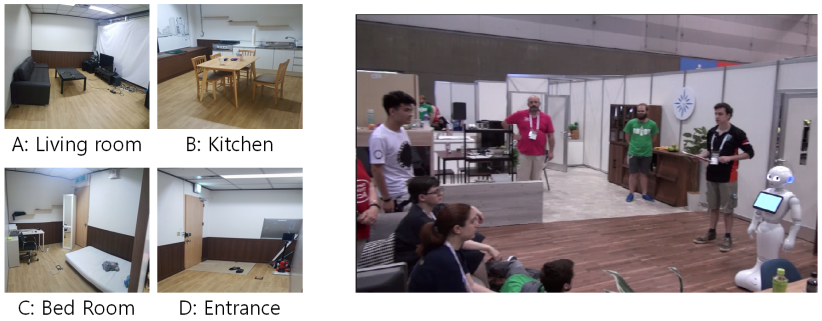

Fig. 5. Left : Our home-like environment. Right : Our team in RoboCup@Home 2017 SSPL

8 Implementation Details

Details about the name of models we used for our system and the link to the open sourced softwares can be found in Appendix

9 Conclusion

In conclusion, we propose a deep learning based service robot system for innovation of mobile home robots. We expect to observe the performance of our research outputs in the realistic cognitive tasks of RoboCup@Home 2018 SSPL competition. We are also looking forward to see our system's limit by exploiting it in the complex and unpredictable arena environment. A wider range of related studies can be found on our website, such as dialogue and motion generation.

References

[1] Ejaz Ahmed, Michael Jones, and Tim K Marks. "An improved deep learning architecture for person re-identification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3908–3916.

[2] Zhe Cao et al. "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: *CVPR*. 2017.

[3] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. "Densecap: Fully convolutional localization networks for dense captioning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4565–4574.

[4] Jin-Hwa Kim et al. "Multimodal residual learning for visual qa". In: *Advances in Neural Information Processing Systems*. 2016, pp. 361–369.

[5] Kyung-Min Kim et al. "Pororobot: A deep learning robot that plays video Q&A games". In: *AAAI 2015 Fall Symposium on AI for Human-Robot Interaction (AI-HRI 2015)*. 2015.

[6] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[7] Chung-Yeon Lee et al. "Schedulebot: A Home Robot Learning and Acting Schedule Adaptively via Dynamic Environments". In: (2016).

[8] Sang-Woo Lee et al. "Dual-Memory Deep Learning Architectures for Lifelong Learning of Everyday Human Behaviors." In: *IJCAI*. 2016, pp. 1669–1675.

[9] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[10] Joseph Redmon and Ali Farhadi. "YOLO9000: better, faster, stronger". In: *arXiv preprint arXiv:1612.08242* (2016).

Appendix

Team Name : AUPAIR

Contact Information : Chung-Yeon Lee (cylee@bi.snu.ac.kr)

Website : https://bi.snu.ac.kr/Robocup/2018/index.html

Team Members : Chung-Yeon Lee, Kibeom Kim, Sungjae Cho, Eun-Chan Lee, Hyun-Kyu Lee, Joo-Hyun Cho, Sou-Hwan Kim, Joon-Ha Chun, Eui-Joon Jung, Hyun-Do Lee, and Byoung-Tak Zhang

Hardware

Robot Platform : SoftBank Pepper (see Section 3.1)

External Computing Devices : GPU server with Intel i7-6700k processor 3.4 GHz, 32GB RAM, Nvidia TitanX Paskal GPU (see Section 3.2)

Software

Purpose	Name
OS	Ubuntu14.04
Robot control	ROS indigo[1]
Robot control	NaoQI [2]
SLAM & Navigation	ROS Navigation Stack [3]
Object Detection	YOLOv2[4]
Person Identification	Improved Deep Learning Architecture[5]
Pose Estimation	OpenPose[6]
Image Captioning	DenseCap[7]
Integrated Perception	IPSRO[8]
Speech Recognition	Google Cloud API[9]

[1] http://wiki.ros.org/indigo
[2] http://doc.aldebaran.com/2-5
[3] http://wiki.ros.org/navigation
[4] https://pjreddie.com/darknet/yolo/
[5] https://github.com/Ning-Ding/Implementation-CVPR2015-CNN-for-ReID
[6] https://github.com/CMU-Perceptual-Computing-Lab/openpose
[7] https://github.com/jcjohnson/densecap
[8] https://github.com/gliese581gg/IPSRO
[9] https://cloud.google.com/speech/